基于 ReliefF 和改进乌鸦搜索优化的并行入侵检测方法 *

马　超

(深圳信息职业技术学院 数字媒体学院，广东 深圳 518172)

摘　要：网络数据量的增加导致计算复杂度和时间复杂度增加，为提高网络入侵检测的检测精度与速度，提出一种新的入侵检测方法 RICSA-KELM。首先采用 ReliefF 过滤法除去无关特征和噪声，降低特征维数；然后采用封装法基于改进乌鸦搜索算法(ICSA)进行最优特征子集选择，并同步实现核极限学习机(KELM)分类器的参数优化。设计的线性加权目标函数在考虑最大分类精度的同时，尽可能减少误报率以及特征子集数量。此外提出基于多核平台的多线程并行计算方法，进一步优化模型运算方式，提高了计算效率。实验采用 KDD99 和 UNSW-NB15 测数据集对 RICSA-KELM 性能进行测试和分析。实验结果表明，提出的模型优于 SVM、ELM、KNN 等方法，检测准确率高、检测效率快、误报率低，是一种有效的网络入侵检测方法。

关键词：乌鸦搜索算法；入侵检测；并行计算；核极限学习机；ReliefF

中图分类号：TP183　　doi: 10.3969/j.issn.1001-3695.2018.06.0309

Network intrusion based on ReliefF and improved crow search optimization parallel method

Ma Chao

(*College of Digital Media*, *Shenzhen Institute of Information Technology*, *Shenzhen Guangdong* 518172, *China*)

Abstract: The increase of network data leads to the increase of computation complexity and time cost, in order to further improve the detection accuracy and efficiency of network intrusion detection, a novel algorithm RICSA-KELM was proposed. Firstly, filter method ReliefF used to remove the irrelevant features and noises, and reduced the feature dimension. Secondly, wrapper method used to select optimal feature subset which based on improved crow search algorithm (ICSA) , and optimized parameters of kernel extreme learning machine (KELM) . Moreover, a linear-weighted multi-objective function designed to take into account the average accuracy rate, false alarm rate and the subset of feature selection, it helped to improve the accuracy of the algorithm. At last, RICSA-KELM implemented in parallel on multi-core processor by using OpenMP to speed up the search and optimization process. Experiment on KDD99 dataset and UNSW-NB15 dataset, by means of the experimental analysis and comparison with ELM, SVM and KNN, the proposed method not only improves the detection accuracy and detection efficiency, but also achieves lower false positive rate, it proves that the validity of the proposed method.

Key words: crow search algorithm; intrusion detection; parallel computing; kernel extreme learning machine; ReliefF

0　引言

随着互联网技术的日益普及，网络攻击手段多样化，攻击数量和危害程序也呈上升趋势，传统防火墙和数据加密等防范手段已不能满足现代网络安全的需求。网络入侵检测作为一种主动防御技术，它能从网络收集和分析系统的安全数据，提取出系统的各种行为模式以及行为特征，实时地保障网络安全，并及时发出警报，因此它已成为信息安全领域的研究热点[1]。

入侵检测通常包括误用入侵检测和异常入侵检测。前者检测率高，但不能发现变异的入侵行为，而后者则可以发现新的入侵行为，所以当前主要针对异常入侵检测进行研究。在模式识别领域中网络入侵检测实质上是一个多分类问题，主要包括特征选择和分类器设计两部分。近年来随着人工智能技术的快速发展，越来越多的研究人员尝试将人工智能技术用于网络入侵检测[2]。例如 2016 年 Bamakan 等人[3]改进随时间变化的混沌粒子群(PSO)算法，并分别结合多准则线性规划(MCLP)和支持向量机(SVM)进行入侵检测研究，在 KDD 数据集上验证了该方法的有效性；同年李从等人[4]提出一种融合 FAST 特征选择与

收稿日期：2018-06-19；**修回日期：**2018-07-31　**基金项目：**国家自然科学基金青年基金资助项目（61303113）；广东省自然科学基金资助项目（2016A030310072）；深圳市科技计划项目（KJYY20170724152553858）；广东省教育厅重点平台及科研项目特色创新类项目（2017GWTSCX040）；深圳市 2017 年度规划课题（ybzz17011，ybzz17009，zdzz17005）；校级科研课题（QN201716）

作者简介：马超（1983-），男，辽宁葫芦岛人，讲师，博士，主要研究方向为数据挖掘、机器学习、模式识别（billmach@163.com）.

自适应二进制量子引力搜索支持向量机的 FAST-ABQGSA-SVM 网络入侵检测算法，在 KDD CUP99 上实验证明该算法较具有较好的鲁棒性和学习精度；华辉有等人[5]提出了融合 Kmeans 和 KNN 的网络入侵检测算法 Cluster-KNN，将聚类和分类结合分别进行离线预处理和在线分类两阶段，证明了该方法在准确率、误报率和漏报率方面与其他同领域入侵检测方法相比也具有一定优势；2017 年 Akashdeep 等人[6]提出基于信息增益 IG 进行特征排序并结合神经网络 ANN 分类器的入侵检测系统，在 KDD-99 数据集上得到了很好的结果；同年 Wang 等人[7]提出基于增强特征和 SVM 的入侵检测框架，基于密度比率通过将原始特征转换到更高质量的特征空间，算法在 KDD 数据集上得到了鲁棒性更强的结果；Raman 等人[8]提出基于超图遗传算法(GA)同步优化 SVM 参数和特征选择进行入侵检测，算法在 KDD 数据集上进行了测试得到了良好的结果；Aburomman 等人[9]设计了基于差分进化算法优化加权 SVM 多分类器模型用于入侵检测，在 NSL-KDD 数据集上取得了很好的分类结果；2018 年 Hajisalem 等人[10]提出人工蜂群(ABC)和人工鱼群(AFS)的混合分类方法用于入侵检测研究，并在 KDD 和 UNSW-NB15 数据集上取得了很高的精度和误报率；同年 Chiba 等人[11]提出优化 BPNN 的方法用于异常入侵检测研究，取得了较好的效果。

从这些研究中可以发现，基于人工智能技术的网络入侵检测模型和方法已得到广泛的关注和研究，其中基于 SVM 和神经网络的模型使用最广泛，取得了较好的效果，但仍存在以下四个问题：

a)对于神经网络方法，其权重参数优化多采用基于梯度下降的方法，易陷入局部极小值，算法训练时间过长，需要多次迭代才能收敛，学习速度很慢，而 SVM 模型的核函数和参数选择问题，这对入侵检测的结果有重要且直接的影响，然而对于参数的选择尚未有统一的标准和理论指导，目前研究人员大多采用群智能算法对参数迭代寻优，搜索易陷入局部极值而找不到全局最优解。

b)特征选择和分类器优化两者同样重要，通常是分开独立进行，之前工作较少考虑两者的相关性。

c)虽然 SVM 方法可以处理二分类问题得到了很好的效果，但不能直接用于入侵检测这种多分类问题，需要采用“一对一”或“一对多”的复杂方式构建成多分类器。

d)现有的大多数方法都是基于串行运行方法的研究，计算效率低下，未能考虑在并行条件下进一步提高网络入侵检测效率。

基于之前分析，为了克服以上缺点，进一步提高网络入侵的检测准确率和检测效率；同时考虑到特征选择和分类器优化是具有相关性，对检测精度具有同样重要的影响，提出 RICSA-KELM 模型，用于网络入侵检测。模型采用过滤和封装结合的混合特征选择方法，先利用 ReliefF 进行特征降维，去除不相关特征和噪声；然后提出将改进的乌鸦搜索算法(improved crow search algorithm, ICSA)与核极限学习机(KELM)分类模型结合，模型中 ICSA 算法由连续 CSA 算法和离散 CSA 算法两部分构成，连续 CSA 算法用于自动调节 KELM 参数，离散 CSA 算法用于最优特征子集选择，同时引入混沌运算，增加种群多样性，更好地平衡了全局搜索和局部搜索。使用基于多核平台的 OpenMP 多线程并行方法提高了 ICSA 算法性能，充分利用 CPU 资源改善算法性能，进一步提高了算法的计算效率。

1　ReliefF 算法

ReliefF 算法由 Konomenko 等人提出的基于 Relief 算法扩展提出的一种能用于多分类问题的过滤特征选择方法[12]，它通过计算同类与不同类间的相邻样本来评估样本相关性和冗余度。

ReliefF 算法从原始数据样本集中随机选出样本子集 p，然后从 p 同类样本集中选出 s 个最近邻样本，并从与 p 不同类的样本集中选出 s 个最近邻样本，计算得到每个特征权重值并依次更新。重复上述过程，直到计算得到这些样本中各特征与类别的相关度。然后将特征根据其特征权重值进行降序排列，通过给定阈值来选择部分特征集合。即当特征权重值大于给定阈值的特征用于构成新的特征子集，若小于给定阈值则去除掉该特征。ReliefF 算法实现如下：

输入：p 个样本实例及其对应的特征属性。

输出：特征权值向量 w。

a)初始化 w=0。

b)从 p 个样本中选择一个，并从 p 同类和不同类中分别选出 s 个最近邻样本，并计算特征权重值。权重值 w 计算公式如下：

$$w[I]=w[I]-\sum_{j=1}^{s}diff(I,P_i,H_j)/ms+$$
$$\sum_{C\neq class(P_i)}[\frac{P(C)}{1-P(class(P_i))}\sum_{j=1}^{s}diff(I,P_i,M_j(C))]/ms \tag{1}$$

其中：m 为样本抽样次数；diff(I,P_j,H_j)函数计算两样本实例关于特征 I 的距离；$M_j(C)$为不同类的第 j 个最近邻样本；$class(p_i)$为 p_i 样本的类别。

ReliefF 算法具有易扩展、有效性强、稳定性好、计算效率高等优点，能快速处理大量数据和噪声数据，是一种较好的过滤评估算法。由于 ReliefF 算法的特征评价过程中考虑特征之间的相关性，所以能够较好地去除无关特征。

2　改进的乌鸦搜索算法(ICSA)

乌鸦搜索算法 CSA 是由 Askarzadeh[13]于 2016 年提出的一种新的群智能优化算法，它模拟的是自然界中乌鸦的智能觅食行为。乌鸦算法简单易实现、鲁棒性强，涉及的需要调节的参数较少，在网络优化等领域有一定的应用研究。乌鸦是群居生活的具有很高智慧的鸟类，它们找到食物后通常将多余的食物藏匿起来，藏匿位置称为记忆值(memory)，在需要时取出;当前

能跟踪其他乌鸦，窃取其他乌鸦的食物，而被跟踪的乌鸦能以一定的感知概率(awareness probability，*AP*)保护自己的食物防止被窃。

2.1 连续 CSA 算法

在求解最优问题时，假定 *N* 只乌鸦随机分布在 *n* 维搜索空间中，$x^{i,t}=[x_1^{i,t}, x_2^{i,t},\dots, x_n^{i,t}]$($i$=1,2,…,$N$; t=1,2,…,$Maxiter$)表示第 *i* 只乌鸦在第 *t* 次迭代时的位置。$M^{i,t}$ 表示乌鸦 *i* 在第 *t* 次迭代时隐藏食物的记忆值，即最优位置。$AP^{i,t}$ 表示乌鸦 *i* 在第 *t* 次迭代时的感知概率 *AP*，$fl^{i,t}$ 表示乌鸦 *i* 在第 *t* 次迭代时的飞行长度；

对乌鸦搜索算法进行初始化控制参数设置，所述初始化控制参数包括种群群体数量 *M*、感知概率 *AP*、飞行长度 *fl* 以及最大迭代次数 *Maxiter*；

传统乌鸦搜索算法是随机初始化位置，公式如下：

$$x^{i,t}=rand\cdot(x_{\max}-x_{\min})+x_{\min} \tag{2}$$

其中：$x^{i,t}$ 为乌鸦随机产生的位置；$x_{\max}$ 为 *x* 的最大值；$x_{\min}$ 为 *x* 的最小值；*rand* 为[0,1]区间随机生成数。

但是随即初始化导致个体的质量无法保证，解群中有一部分远离最优解的位置，如果初始解群较好，将会有助于求解效率与解的量；如果不好，则会影响求解效率，增加了不确定性。而一个好的初始化种群能够确保算法更快地收敛。本文将混沌算法优化乌鸦搜索来解决上述问题。混沌运动一种貌似随机的运动，是在确定性非线性系统中自然出现的类随机行为，它具有确定性过程同时也兼具随机性[14]。混沌运动这种非线性系统所特有的一种形式，可以使得算法能够跳出局部最优的同时寻找全局最优解。因此本文采用混沌映射函数 Logistics 对乌鸦位置进行初始化：

$$X_{n+1}=\mu\cdot X_n\cdot(1-X_n)\quad \mu\in[0,4], X_n\in(0,1) \tag{3}$$

其中：参数 μ 用于控制混沌程度。

在第 *t* 次迭代时，乌鸦 *i* 随机选择一只乌鸦 *j* 跟踪以偷窃对方的食物。算法包括全局搜索和局部搜索两部分。通过感知概率 *AP* 进行动态调整以达到全局搜索和局部搜索的平衡状态。当随机生成数大于等于乌鸦感知概率 *AP* 时，即乌鸦 *j* 知道乌鸦 *i* 跟踪它，会把乌鸦 *i* 带到任意位置；反之，当随机生成数小于 *AP* 时，即乌鸦 *j* 不知道乌鸦 *i* 跟踪它，则乌鸦 *i* 向乌鸦 *j* 的最优位置移动。由于乌鸦位置的更新影响着最优解和收敛速度，引入混沌算法进一步优化乌鸦搜索位置的更新。位置更新的表达式如下：

$$x^{i,t+1}=\begin{cases}x^{i,t}+w_i\cdot r_i\cdot fl^{i,t}\cdot(m^{j,t}-x^{i,t}), & \text{if}\quad w_z\ge AP^{j,t}\\ rand\cdot(x_{\max}-x_{\min})+x_{\min}, & else\end{cases} \tag{4}$$

其中：w_i 表示在第 i 代时得到的混沌映射值；w_z 表示在第 *z* 代得到的混沌映射值；$AP^{i,t}$ 表示乌鸦 *j* 在 *t* 代时的感知概率；r_i 和 r_j 是[0,1]区间均匀分布的随机数。

由式(4)可知，通过混合函数的引入进一步平衡算法全局搜索和局部搜索，对全局搜索和局部搜索进行更加灵活地动态扰动，在前期 w_i 值较大，确保全局搜索占较大权重，提高种群搜索的多样性；到迭代后期，w_i 值变小，使得局部搜索权重加大，加速算法收敛。

当乌鸦 *i* 的位置发生改变，则更新记忆值表达式如下：

$$M^{i,t+1}=\begin{cases}x^{i,t+1}, & \text{if } f(x^{i,t+1})>f(M^{i,t})\\ M^{i,t}, & else\end{cases} \tag{5}$$

其中：$M^{i,t}$ 表示乌鸦记忆值；$f(M^{i,t})$表示适应度值。

2.2 离散 CSA 算法

为了更有效地处理实际问题，Sayed 等人[15]又提出离散 CSA 算法用于特征选择。其中种群个体的每一维和最优位置均为 0 或 1，引入映射函数 $S(x)$ 将连续空间的值转换到离散空间[0, 1]：

$$M^{i,t+1}=\begin{cases}1, & \text{if } f(S(M^{i,t+1}))\ge rand()\\ 0, & else\end{cases} \tag{6}$$

图 1 为 ICSA 算法的整体流程。

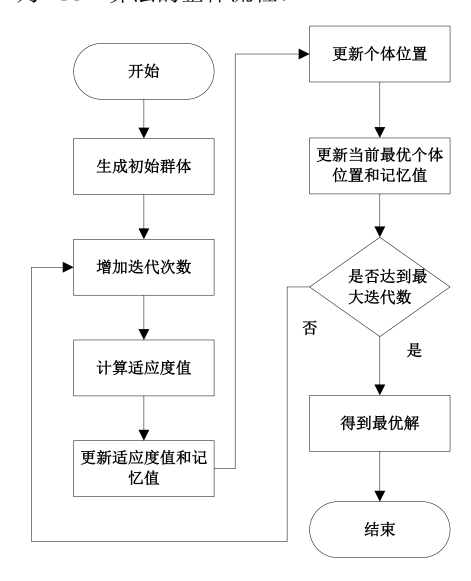


图 1 ICSA 算法的流程

3 RICSA-EKELM 模型

在本章将详细介绍 RICSA-KELM 分类模型。该模型先利用 ReliefF 过滤方法除去不相关特征和噪声，然后在并行环境下，自适应确定 KELM 参数并找出最具区分力的特征子集，采用 ICSA 算法同步进行参数优化和特征选择。在提出的 RICSA-KELM 模型中，将同时考虑三个子目标函数设计适应度函数，即 KELM 模型所得到的 ACC 值、误报率、特征子集的大小。模型整体流程如图 2 所示。下面分别介绍串行算法，接着实现并行算法。注意，本阶段实施的前提是已经基于 ReliefF 过滤特征选择方法对原始样本集进行了特征降维处理，去除了不相关特征和噪声之后的特征集合。

3.1 串行模型

RICSA-KELM 模型的运行过程包括三个阶段：第一阶段，利用 ICSA 算法通过迭代搜索寻找最优特征集合和 KELM 参数组合；第二阶段，利用第一阶段提供的最优特征集合和最优参数组合在不同训练数据集上进行训练，得到 RICSA-KELM 分类器；第三阶段，利用训练好的分类器在测试数据集上进行测试，线性加权多目标函数同时考虑了分类精度 ACC、误报率以

及特征个数三个子目标函数。主要步骤为：

a)对解进行编码，编码长度为 n+2 维，其中前 n 维由 0 和 1 二进制数组成，1 表示选中该特征，0 表示该特征未被选中，最后两维分别表示 KELM 的参数 C 和 γ 两个连续值，解编码形式为 $X=[0,1,\ldots,1,0,C,\gamma]$。

b)进行种群初始化，设定相关参数，包括种群大小、最大迭代次数、感知概率 AP、飞行长度 fl。

c)利用步骤 b)初始化个体解码得到的特征集合和参数在 KELM 上训练。

d)ACC 越高，误报率越低，特征子集越小,可得到更高的适应度值，因此对三者综合考虑设计了线性加权多目标函数，计算公式如下：

$$f_1=ACC=\frac{\sum_{i=1}^{K}accuracy_i}{K} \tag{7}$$

$$f_2=1-FA \tag{8}$$

$$f_3=(1-\sum_{j=1}^{n}m_j\Big/n) \tag{9}$$

$$F=\mu\cdot f_1+\eta\cdot f_2+\sigma\cdot f_3 \tag{10}$$

其中:f_1表示 KELM 的 K 折交叉验证(K-fold cross validation, K-fold CV)[16]所得的 ACC 值；f_2中 FA 表示误报率(false alarm rate，FA)；f_3中 m 表示特征值，n 为特征总数。F 中 μ、η 和 σ 为常量值；μ 为 KELM 准确率的权重；η 为误报率权重；σ 是所选特征集合的权重，$\mu+\eta+\sigma=1$。权重可调整到一个恰当的值，这取决于各子目标函数对评估结果贡献大小。由于分类性能更依赖准确率和误报率，所以根据实验多次尝试，μ、η 和 σ 分别设为 0.5。0.3 和 0.2。

e)增加迭代次数 $t=t+1$。

f)根据式(4)和(5)更新种群个体的位置和记忆值。

g)利用步骤 f)得到更新解，进行解码得到的特征集合和参数在 KELM 上训练，根据式(7~10)计算个体适应度值。

h)记录当前种群的最优解，若当前适应度值大于存储的最优适应度值，则更新适应度值为当前值，否则保持存储的适应度值不变。

i)若达到最大种群数，转到步骤 j)运行，否则算法转到步骤 f)运行。

j)比较当前适应度值和全局最优适应度值，若当前值大于存储的全局最优适应度值，更新为当前值，否则保持历史最优适应度值不变。

k)若达到最大迭代次数，转到步骤 l)，否则转到步骤 e)继续进行迭代寻优。

l)输出全局最优解，通过解码得到最优特征子集和最优参数组合(C,γ)。

m)利用最优特征子集和参数组合并结合训练集在 KELM 上训练，得到最优分类器模型。

n)在测试集上测试并得到最终分类结果。

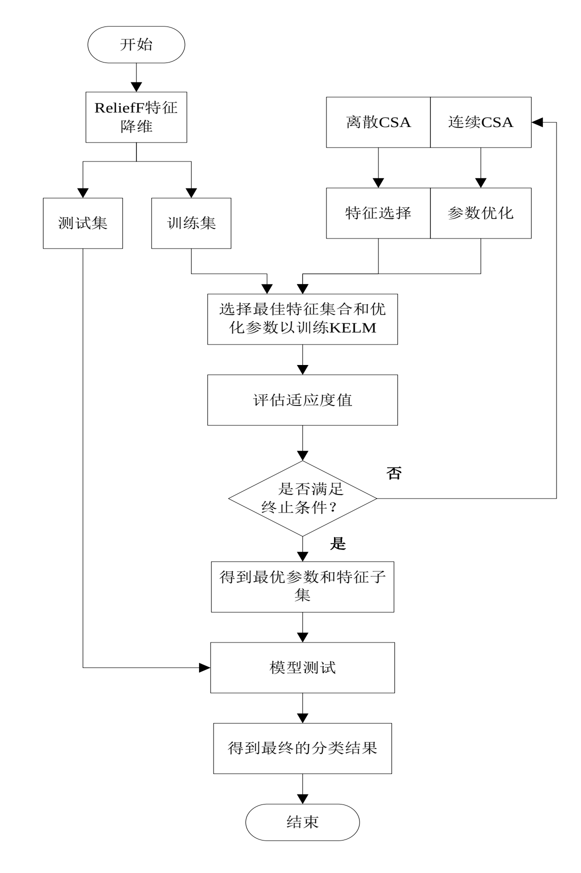


图 2　RICSA-KELM 模型的总体流程

3.2　并行模型

对于许多复杂优化问题，ICSA 算法需要多次更新才能保证找到最优解。ICSA 算法的初始解生成、适应度计算、记忆值更新等在算法中比较耗时，并且它们是相互独立的，所以 ICSA 算法具有天然的并行性。为充分发挥 ICSA 算法并行性，提高算法效率，提出基于多核处理器利用 OpenMP[17]来实现并行模型。多核平台的整体框架由三层组成：

a)ICSA-KELM 模型。该层由一系列种群个体组成，并行算法控制整个 CSA 迭代过程，每个个体独立参与整个运算过程。

b)OpenMP 平台。该层是为保证实现并行算法的同步，同时建立和操作系统间的通信联系。平台核心组件是调度器，能给操作系统提供作业的调度和分配。

c)多核处理器。作业在该层通过 OpenMP 被系统调用。

并行 RICSA-KELM 算法的伪代码如下：

```
Initialize model parameters
Train KELM;
Calculate the fitness;
while i<max_iteration /* i 为当前迭代次数, max_iteration 为最大迭代次数*/
for each particle
Update position;
Update memory;
Train KELM;
Calculate the fitness;
Calculate fitness_best;
```

```
Calculate memory_best;
end for;
    Calculate fitness_global;
    Calculate memory_global;
    i=i+1;
end while
```

4 实验分析

4.1 数据描述与处理

实验采用的是 KDD99 和 UNSW-NB15 的部分网络入侵检测数据集，其中 KDD99 数据集是使用最多的经典数据集，很多入侵检测方法有基于此数据集，UNSW-NB15 数据集是澳大利亚网络安全中心 ACCS 研究小组于 2015 年创建[18]。数据集信息分别如表 1 和 2 所示。限于空间未给出 UNSW-NB15 数据集的具体特征描述信息。

表 1 KDD99 数据集四种异常类型数据信息

类别	数量	攻击类型
Dos	10,000	Ipsweep,nmap,portsweep,satan
Probe	4,107	Back,land,Neptune,pod,smurf,teardrop
R2L	1,126	Rootkit,perl,loadmodule, buffer_overflow
U2R	52	ftp_write,multihop,warezclient,ph, warezmaster,imap,guess_passwd,spy

表 2 UNSW-NB15 数据集数量及分布比例信息

类别	数量	分布比例/%
Backdoor	1,746	2.87
Analysis	2,000	3.29
Fuzzers	10,000	16.46
Shellcode	1,133	1.86
Reconnaissance	10,491	17.26
Exploits	13,000	21.39
Dos	12,264	20.18
Worms	130	0.21
Generic	10,000	16.46

为缩小特征值之间大小差异，采用数据归一化对数据集进行预处理，将所有特征值映射到[0,1]之间，避免较大数量级数据对较小数量级的数据造成干扰，保证结果的有效性。计算公式如下：

$$x_i = (x_i - x_{min}) / (x_{max} - x_{min}) \tag{11}$$

其中:x_{min} 为数据集中最小值;x_{max} 为数据集中最大值。此外，在实验过程中为避免过拟合和欠拟合现象的发生，使得结果更具说服力，采用双层交叉验证方法[19]，内层 10 折交叉验证确定最优特征子集和参数，外层 5 折交叉验证评估 KELM 的分类性能。由于运行一次交叉验证不能保证结果的公正性，数据是随机抽样分割的，每次得到的训练集和测试集不会完全一样，在实验中运行十次，然后求平均值为最终结果

4.2 实验设置

提出的 RICSA-KELM 算法是 MATLAB 2014b 开发环境下设计实现的。KELM 采用的 ELM 工具包，硬件平台配置为 Intel 四核处理器，主频 3.2 GB，16 GB 内存，64 位 Windows8 操作系统。

在接下来 RICSA-KELM 模型的训练和相关算法比较过程中，模型的详细参数设置如下：粒子初始位置和速度均设为[0,1]之间的随机数，种群数量为 20，最大迭代次数为 100。为了公平比较，ICSA-KELM 模型的设置与 PSO-SVM 相同。KELM 和 SVM 模型中 C 和 γ 的搜索范围为 $C\in\{2^{-10},\ldots,2^{15}\}$和 $\gamma\in\{2^{-15},\ldots,2^{5}\}$。

4.3 实验结果分析

为了验证提出的模型有效性，实验首先给出分类模型 RICSA-KELM 在两个数据集上与其他三种模型(KELM、SVM、LSSVM)在 KDD99 和 UNSW-NB15 数据上的分类性能比较，如表 3 和 4 所示。其中 RICSA-KELM 模型利用 ICSA 算法对 KELM 模型进行参数优化和特征选择。对 SVM 和 LSSVM 算法是利用网格计算方法对参数进行优化。同时表中给出了其他三种模型在该数据集上的准确率 ACC 和误报率。ACC 越高，误报率越低说明性能越好。从结果可以看出，在 KDD99 数据集上这四种分类方法中提出的模型取得了最高的平均分类准确率 95.88%，明显高于其他三种分类方法，之后依次排列的分类器是 KELM、LSSVM 和 SVM。提出的模型在分类精度上比其他三种模型分别提高了 1.04%、3.64%和 3.46%，同时提出模型具有最低的误报率为 1.28%。在 UNSW-NB15 数据上本方法也取得了较好的结果。

表 3 四种模型在 KDD99 数据集上分类性能比较

	提出方法		KELM[20]		SVM[21]		LSSVM[22]	
	ACC (%)	FA (%)	ACC (%)	FA (%)	ACC (%)	FA (%)	ACC (%)	FA (%)
1#	95.61	1.34	92.81	2.93	91.87	3.02	91.21	3.25
2#	96.02	1.12	93.72	2.82	92.69	2.34	92.69	3.69
3#	95.82	1.36	93.77	2.56	92.38	2.39	93.38	3.26
4#	96.74	1.50	94.29	2.07	92.14	2.63	92.32	3.54
5#	95.20	1.09	94.51	2.21	92.12	2.82	92.52	3.37
均值	**95.88**	**1.28**	93.82	2.52	92.24	2.64	92.42	3.42

表 5 和 6 给出了经特征选择和未经特征选择的 RICSA-KELM 模型在两个数据集上的测试结果。从结果可见，在 KDD99 数据集上，经特征选择的模型比未经特征选择的模型在性能上提高了 ACC 1.72%；在 UNSW-NB15 数据集上，经特征选择的模型比未经特征选择的模型在性能上提高了 ACC1.83%，这体现了 RICSA-KELM 良好的性能，其优越性是由于设计改进的 ICSA 算法辅助 KELM 通过自动调节优化参数得到最优的分类性能。为了统计上验证提出的方法分类性能效果是否显著，在测试中进行 Wilcoxon 符号秩检验(Wilcoxon

signed ranks test)[23]，置信区间为 0.95。从表中 t 检验结果可以看出，RICSA-KELM 模型与未经特征选择的模型比较，在四个性能指标上均具有显著性差异，P 值均小于 0.05，这表明在两个数据集上，提出的模型在分类性能上均有较大的提高。另外从表中还可发现，提出模型的均方差相对较小，说明该模型具备很好的稳定性。

表 4　四种模型在 UNSW-NB15 数据集上分类性能比较

	提出方法		KELM		SVM		LSSVM	
	ACC (%)	FA (%)	ACC (%)	FA (%)	ACC (%)	FA (%)	ACC (%)	FA (%)
1#	93.46	2.13	90.82	3.56	90.20	3.86	88.29	4.51
2#	93.02	2.24	90.20	3.87	89.56	4.53	90.12	4.39
3#	93.52	2.10	92.31	3.21	91.21	3.79	91.63	3.87
4#	95.01	1.85	91.87	3.55	88.12	4.67	92.32	3.62
5#	93.81	2.29	91.23	3.61	90.95	3.42	91.05	3.37
均值	**93.76**	**2.12**	91.28	3.56	90.01	4.05	90.68	3.95

表 5　KDD99 数据集原始空间和特征选择后的分类比较

性能比较	RICSA-KELM 在原始空间	RICSA-KELM 经过特征选择	T 检验 P 值
ACC(%)	94.16±1.34	95.88±0.94	0.043
FA(%)	2.23	1.28	

表 6　UNSW-NB15 数据集原始空间和特征选择后的分类比较

性能比较	RICSA-KELM 在原始空间	RICSA-KELM 经过特征选择	T 检验 P 值
ACC(%)	91.93±2.77	93.76±1.23	0.036
FA(%)	3.67	2.12	

表 7 给出 RICSA-KELM 在 KDD99 数据集上的详细测试结果。可以看到每一折 C 和 γ 值都不同，通过寻找最优(C,γ)组合 KELM 在每一折样本上都取得良好的分类效果。这是由于 ICSA 算法在迭代过程中自适应调整参数组合，并能根据数据样本的分布进行演化。提出的 RICSA-KELM 算法不仅实现了 KELM 的参数优化，也同步实现了特征选择机制。表 8 给出的是模型与相近模型的分类结果比较。

表 7　模型在 KDD99 数据集上 10-CV 每一折上得到的结果

10 折 CV	C	γ	ACC(%)	误报率(%)
1#	122.367	3.789	96.31	1.15
2#	78.067	2.751	95.70	1.68
3#	57.021	15.691	96.42	0.89
4#	32.554	3.052	95.14	1.14
5#	79.133	0.138	95.68	0.92
6#	37.121	3.125	96.22	0.81
7#	11.654	3.125	96.56	0.73
8#	0.876	0.138	95.90	0.82
9#	10.796	8.132	95.17	1.33
10#	37.781	3.125	96.35	0.82
均值			**95.94**	

表 8　提出的模型与相近模型在 KDD99 数据集分类结果对比

性能比较	ACC(%)	FA(%)
提出方法	95.88±0.94	1.28
CSA-KELM	94.49±1.56	2.25
CSA-SVM	91.32±3.43	3.89

为了进行充分比较，分别实现了基于 PSO 优化 KELM 算法(PSO-KELM)和基于 GA 优化 KELM 的算法(GA-KELM)，比较结果如表 9 所示。从结果可知，提出的模型在 ACC 性能指标上高于 PSO-KELM 和 GA-KELM 模型，而且具有较小的误报率，这反映出 ICSA 算法具有比 PSO、GA 更强的搜索优化能力。

表 9　模型与 PSO-KELM 和 GA-KELM 的分类结果对比

性能比较	ACC(%)	FA(%)
提出方法	95.88±0.94	1.21
PSO-KELM	94.12±2.21	2.35
GA-KELM	92.89±3.37	3.53

为了更全面地研究 ICSA 算法的特征选择过程，探究到底是哪些特征参与了 KELM 模型的训练，给出了 RICSA-KELM 在 KDD 99 数据集上 10-CV 所选中的特征集合，如表 10 所示。入侵检测数据集共包含 41 个特征，但并非所有的特征都对分类准确率有帮助，特征选择提高了分类精度，正如表 5 中和 6 所示的结果一样。从图 3 统计特征被选择的频率可知，其中最重要的特征有 F1、F2、F3、F4、F10、F20、F23、F24、F38 和 F39(共 10 个)，这些特征出现的频率要明显高于其他特征(出现频率次数大于等于 7)，分别对应 KDD 99 数据集的特征项分别为 duration(1)、protocol_type(2)、service(3)、flag(4)、hot(10)、num_outbound_cmds(20)、count(23)、srv_count(24)、dst_host_serror_rate(38)、dst_host_srv_serror_rate(39)。进一步研究这些与网络入侵相关的因素，为入侵检测提供更有力依据，从而帮助专家进行及时应对处理。

为了验证并行模型的性能，将并行模型与串行模型进行了比较。表 11 给出了并行模型和串行模型在 KDD99 数据集上的测试结果。从表中可以看到，两个模型在四个性能指标上的结果非常相近，它们的差别在于交叉验证过程数据集的随机选择造成。但在运行时间上串行模型 ICSA-KELM 的平均时间大约是并行模型 RICSA-KELM 的 3 倍。从图 4 也可看到，在每一折过程中并行模型花费的 CPU 时间要远低于串行模型，这表明提出的方法从并行算法获益，弥补传统串行算法耗时过多的问题，极大提高了算法计算效率。

表 10　RICSA-KELM 模型在 KDD99 数据集上选出的特征子集

	选择的特征
1 折	F1, F2, F3, F4, F5, F9, F10, F12, F13, F15, F17, F18, F20, F23, F24, F25, F29, F38, F40
2 折	F1, F3, F4, F6, F7, F8, F11, F13, F15, F16, F17, F18, F19, F20, F23, F24, F25, F30, F38, F39

3 折	F1, F3,F4, F5, F9, F10, F12, F16, F18, F20, F23, F24, F25, F26, F28, F31, F34, F38, F39
4 折	F1,F2, F3, F6, F7, F8, F9, F10, F11, F12, F15, F19, F20, F23, F27, F28, F29, F30, F35, F39
5 折	F1, F2, F3,F4, F6, F7, F9, F10, F13, F16, F20, F22, F23, F24, F25, F27, F33, F34, F35
6 折	F3, F7, F9, F10, F14, F17, F18, F19, F21, F23, F24, F27, F30, F36, F37, F38, F39
7 折	F2, F3, F4, F5, F8, F12, F13, F16, F17, F18, F21, F23, F25, F29, F30, F36, F37, F39
8 折	F1, F2, F3, F4, F5, F7, F8, F16, F19, F20, F24, F27, F29, F36, F38, F39, F40, F41
9 折	F1, F2, F5, F8, F9, F10, F11, F14, F15, F16, F17, F18, F21, F23, F24, F25, F27, F28, F30, F35, F38, F40
10 折	F2, F4, F5, F10, F11, F12, F15, F20, F22,F23, F24, F26, F27, F38, F39, F40

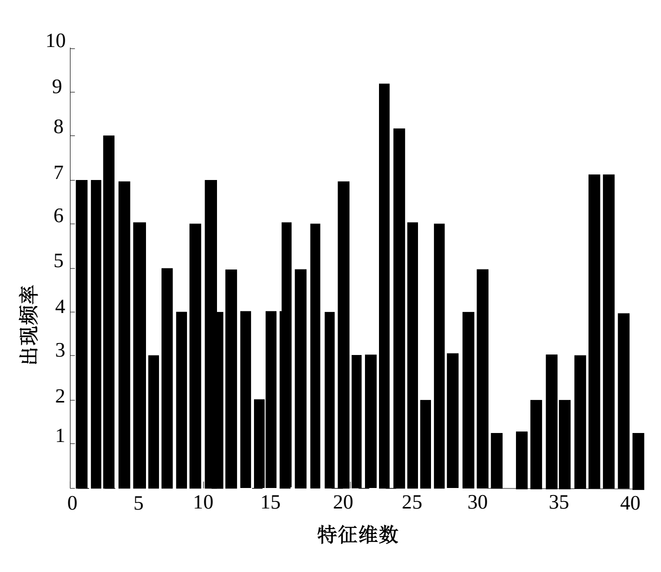


图 3　RICSA-KELM 模型在入侵检测数据集选择特征的频率

为了验证 ICSA 算法全局搜索能力和收敛速度，实验进一步对 ICSA 算法的迭代机制进行研究，给出了 ICSA 算法和 CSA 算法在 KDD99 数据集上 10 折交叉验证中某一折(选的是第 1 折)的最优适应度值变化过程，如图 5 所示。图中给出的是全局最优值的变化过程，将每一次迭代中所有粒子的最优适应度值记录下来。通过观察可知，性能较好的是 ICSA 曲线，从第一次迭代一直到第 100 次迭代逐步演化，ICSA 曲线在初始阶段增长比较迅速，在第 25 次迭代时收敛到最高值，之后适应度值趋于平稳；适应度值较低的是 CSA 曲线，在第 26 次迭代时才收敛到较高值，之后趋于平稳，在 50 代找到最高适应度值，但仍低于 ICSA 曲线，说明 CSA 算法有可能陷入局部最优而未找到全局最优或全局近似最优值。该现象证明了改进的 ICSA 算法比原始 CSA 算法具有更优的全局搜索能力和收敛速度，能迅速收敛到全局最优解，并且在并行环境下高效同步实现了特征选择和参数优化的过程。

表 11　三种模型在 10 折交叉验证下的结果比较

性能比较	ACC(%)	FA(%)	CPU(/s)
并行模型	94.47±1.55	1.53	104.32
串行模型	94.25±1.67	1.32	324.66

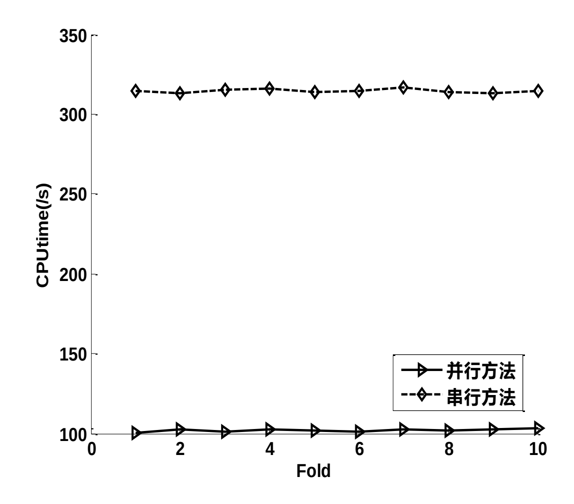


图 4　并行模型和串行模型在 10 折 CV 上的运行时间比较

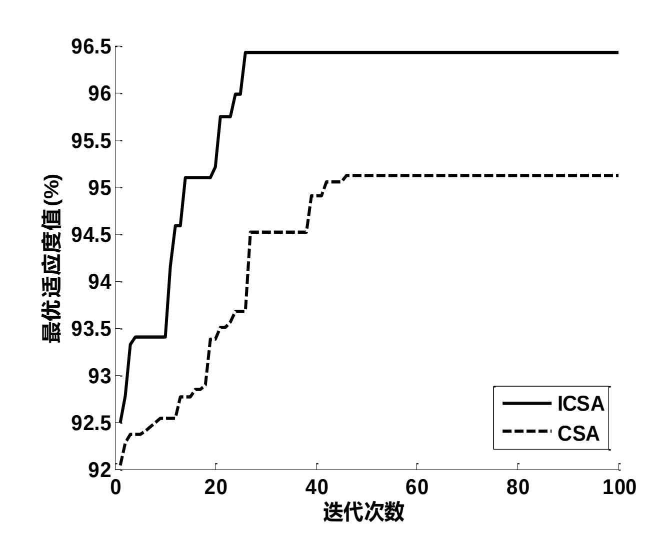


图 5　ICSA 和 CSA 算法在第一折上训练集上最优适应度值

5　结束语

入侵检测是网络信息安全领域中的研究热点和难点，高效且预测率高的入侵检测模型能够更好地处理攻击频繁、复杂的实时网络入侵问题，因此提出了基于过滤和封装混合方法的并行入侵检测模型 RICSA-KELM。在模型中利用 ReliefF 过滤法进行特征降维，剔除大量不相关特征和噪声，然后提出连续型和离散型 ICSA 算法融合到一起同步实现最优特征子集选择和分类器参数优化，不仅实现了特征选择，有效去除网络数据中冗余和不相关的特征，降低了数据维度，提高了算法的效率和分类能力。ICSA 由于加入混沌运算增加了 CSA 群体多样性，同时增强了算法局部搜索能力，提高了 CSA 的全局寻优能力和收敛速度；设计了综合考虑支持分类准确率、误报率和特征个数的线性加权多目标函数；采用 OpenMP 共享存储的并行方式实现了优化分类器的并行计算，极大缩短了 CPU 运行时间，提

高了算法的效率。实验通过在 KDD99 和 UNSW-NB15 数据集上的测试结果表明，与已有方法和相近方法相比，提出的模型取得了较优的参数组合和特征集合，计算效率获得较大提高，并且获得了较好的分类结果，其分类效果优于基于原始 CSA 的 CSA-KELM、PSO-SVM、GA-KELM 和 PSO-KELM 等相近方法，在取得较高的检测准确率，降低了误报率的同时，进一步提高了检测效率，是一种有效的网络入侵检测模型。

当然，还存在许多值得进一步研究的地方。首先，采用的是基于 CSA 算法的优化模型，其他群智能优化算法如 ABC 算法等在该数据集上是否具有更好的表现；其次，采用单一分类器的分类精度有时候容易达到瓶颈，而集成多个分类器的方法效果要高优于单个分类器，采用集成方法进一步提高入侵检测准确率，也是下一步的研究工作。

参考文献：

[1] Muhammad F, Muhammad Sher, Yaxin Bi. Flow-based intrusion detection: techniques and challenges [J]. Computers & Security, 2017, 70: 238-254.

[2] Ashfaq R A R, Wang Xizhao, Huang J Z, *et al.* Fuzziness based semi-supervised learning approach for intrusion detection system [J]. Information Sciences, 2017, 378: 484-497.

[3] Seyed M H B, Huadong W, Tian Y, *et al.* An effective intrusion detection framework based on MCLP//SVM optimized by time-varying chaos particle swarm optimization [J]. Neurocomputing, 2016, 199: 90-102.

[4] 李丛，闫仁武，朱长水等. 融合 FAST 特征选择与 ABQGSA-SVM 的网络入侵检测 [J]. 计算机应用研究，2017，34 (7): 2172-2179. (Li Cong, Yan Renwu, Zhu Changshui. Network intrusion detection based on FAST feature selection and ABQGSA-SVM [J]. Application Research of Computers, 2017, 34 (7): 2172-2179.)

[5] 华辉有，陈启买，刘海等. 一种融合 Kmeans 和 KNN 的网络入侵检测算法 [J]. 计算机科学，2016，43 (3): 158-162. (Hua Huiyou, Chen Qimai, Liu Hai, *et al.* Hybrid K-means with KNN for network intrusion detection algorithm [J]. Computer Science, 2016, 43 (3): 158-162.)

[6] Akashdeep, Ishfaq M, Neeraj K. A feature reduced intrusion detection system using ANN classifier [J]. Expert Systems with Applications, 2017, 88: 249-257.

[7] Huiwen Wang, Jie Gu, Shanshan Wang. An effective intrusion detection framework based on SVM with feature augmentation [J]. Knowledge-Based Systems, 2017, 136: 130-139.

[8] Raman M R G, Nivethitha S, Kannan K. An efficient intrusion detection system based on hypergraph-genetic algorithm for parameter optimization and feature selection in support vector machine [J]. Knowledge-Based Systems, 2017, 134: 1-12.

[9] Abdulla A A, Mamun B. A novel weighted support vector machines multiclass classifier based on different evolution for intrusion detection systems [J]. Information Sciences, 2017, 414: 225-246.

[10] Hajisalem V, Babaie S. A hybrid intrusion detection system based on ABC-AFS algorithm for misuse and anomaly detection [J]. Computer Networks, 2018, 136: 37-50.

[11] Chiba Z, Noureddine A, Khalid M. A novel architecture combined with optimal parameters for back propagation neural networks applied to anomaly network intrusion detection [J]. Computers & Security, 2018, 75: 36-58.

[12] Huang Y, McCullagh P J, Black N D. An optimization of ReliefF for classification in large datasets [J]. Data & Knowledge Engineering, 2009, 68 (11): 1348-1356.

[13] Askarzadeh A. A novel metaheuristic method for solving constrained engineering optimization problems: crow search algorithm [J]. Computers & Structures, 2016, 169: 1-12.

[14] Wang G G, Guo L, Gandomi A H, *et al.* Chaotic krill herd algorithm [J]. Information Sciences, 2014, 274: 17–34.

[15] Sayed G I, Aboul E H, Ahmad T A. Feature selection via a novel chaotic crow search algorithm [J]. Neural Computing & Applications, 2017, 1: 1-18.

[16] Ma Chao, Ouyang J, Chen H L, *et al.* A novel kernel extreme learning machine algorithm based on self-adaptive artificial bee colony optimisation strategy [J]. International Journal of Systems Science, 2016, 47 (6): 1342-1357.

[17] Jaroslav H, Michal L, Stanislav Z. Parallelization of interpolation, solar radiation and water flow simulation modules in GRASS GIS usingOpenMP [J]. Computers & Geosciences, 2017, 107: 20-27.

[18] Moustafa N, Slay J. The evaluation of network anomaly detection systems: statistical analysis of the UNSW-NB15 data set and the comparison with the KDD99 data set [J]. Information Security Journal: A Global Perspective, 2016, 25: 1-3.

[19] Wang Mingjing, Chen Huiling, Yang Bo, *et al.* Toward an optimal kernel extreme learning machine using a chaotic moth-flame optimization strategy with applications in medical diagnoses [J]. Neurocomputing, 2017, 267 (6): 69-84.

[20] Ye Zhifan, Yu Yuanlong. Network intrusion classification based on extreme learning machine [C]// Pric of IEEE International Conference on Information and Automation. 2015: 1642-1647.

[21] Zhou Guangping, Shrestha A. Efficient intrusion detection scheme based on SVM [J]. Journal of Networks, 2013, 8 (9): 2128-2134.

[22] Zhang Hongmei, Gao Haihua, Wang Xingyu. Construct sparse least squares support vector machine for network intrusion detection [J]. Journal of East China University of Science and Technology, 34 (6): 876-881.

[23] Veček N, Črepinšek M, Mernik M. On the influence of the number of algorithms, problems, and independent runs in the comparison of evolutionary algorithms [J]. Applied Soft Computing, 2017, 54 (3): 23-45.